Hardware Modeling using Verilog
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

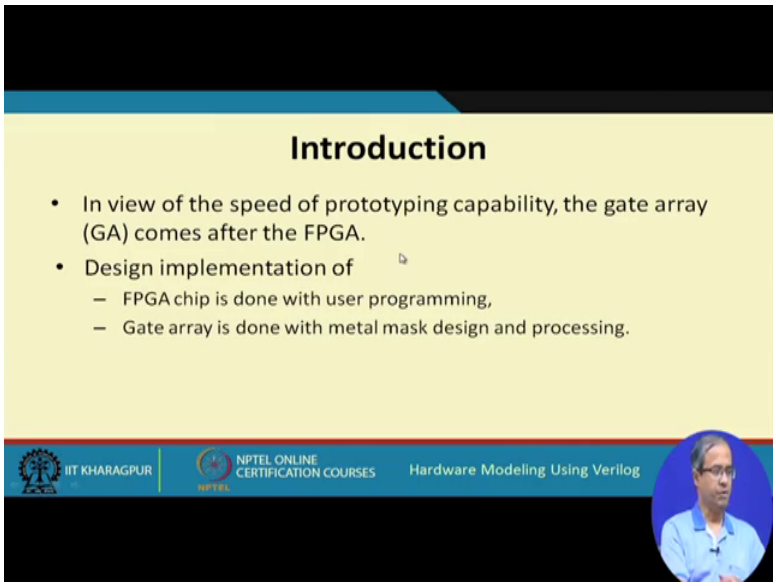
Lecture - 05
VLSI Design Styles (Part 2)

So, in this lecture, we continue with a discussion on design styles in VLSI, if you recall in our last lectures, we had been talking about the FPGA field programmable gate array design style and what are its specific features and rules during the design.

(Refer Slide Time: 00:42)

Now we continue our discussion in that line. So, the first design style that we will be talking about today is something called Gate Array.

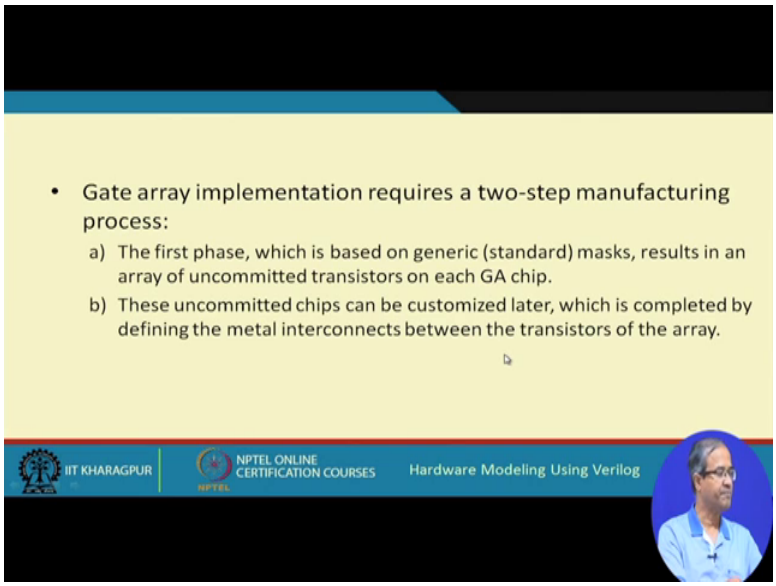
(Refer Slide Time: 00:50)



Now, gate array is something that comes right after FPGA, see the 2 extremes of the spectrum are FPGA on one side and a pure ASIC on the other side. Now you recall what you had said. So, when you want to manufacture an ASIC, we have to send our design to the fabrication facility, they will be carrying out all the steps of fabrication, they will be manufacturing the chip and there will be sending it back to us, but on the other extreme we have this so called field programmable gate array or FPGA where the entire programming and customization can be done by the user in the lab.

So, the turnaround time is extremely fast the FPGA kits as I said will cost you not more than few thousands of rupees; at least the simpler ones. So, it is very cheap very fast and you can very quickly map some design into hardware. Now in that spectrum gate array is something which falls between the 2 extremes, I mean you may say it is a little closer to FPGA. So, let us see the salient features now the main difference for FPGA is that you see FPGA the design customization which we sometimes called programming is entirely done by the user, but for the gate array, it is a 2 step process, user is not doing well here again we are sending it to the fabrication facility; to the fab, but the effort is much less why we shall explain.

(Refer Slide Time: 02:43)



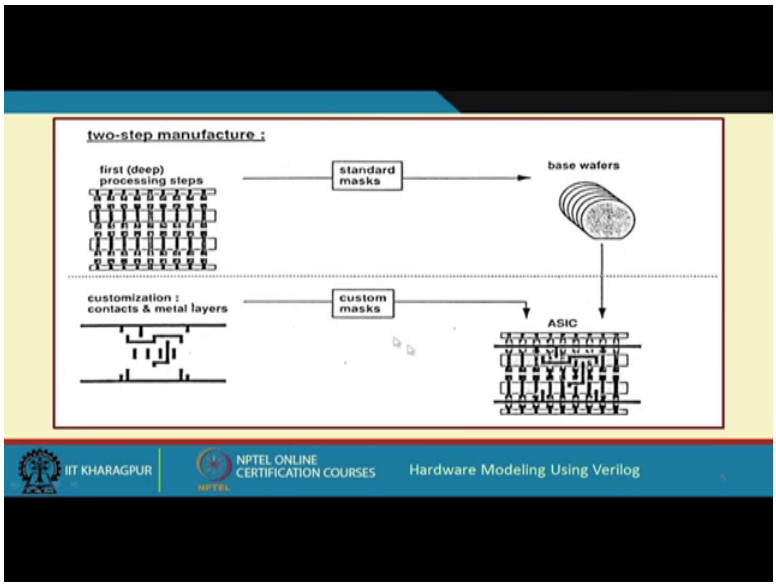
So, for gate array implementation, we actually require a 2 step manufacturing process. So, when we say manufacturing; that means, we are trying to build something.

Now, in a gate array chip which will ultimately implement or realize some functionality, there are 2 steps. First step is independent of the function being realized and the second step is a step of customization. So, you see the first; in the first step, you can complete the first step for all the designs in a means irrespective of what is the function you want implement and once you have done that for specific designs you will have to separately carry out step 2.

So, the first phase which I am saying this is based on some generic masks during fabrication. So, actually what we are doing here we are actually fabricating a large number of transistors on the chip, but what we are doing we are not interconnecting the transistors we are simply fabricating a large number of transistors which is the first phase, Now in this second phase where you are doing the customization, we are completing the metal interconnects we are interconnecting the transistors in a way which will help us in realizing the functionality of a circuit.

These are the 2 phases of manufacturing; first phase is a generic say step which is independent of the design, while this second step is design specific where we carry out the interconnects we interconnect the transistors to form gates.

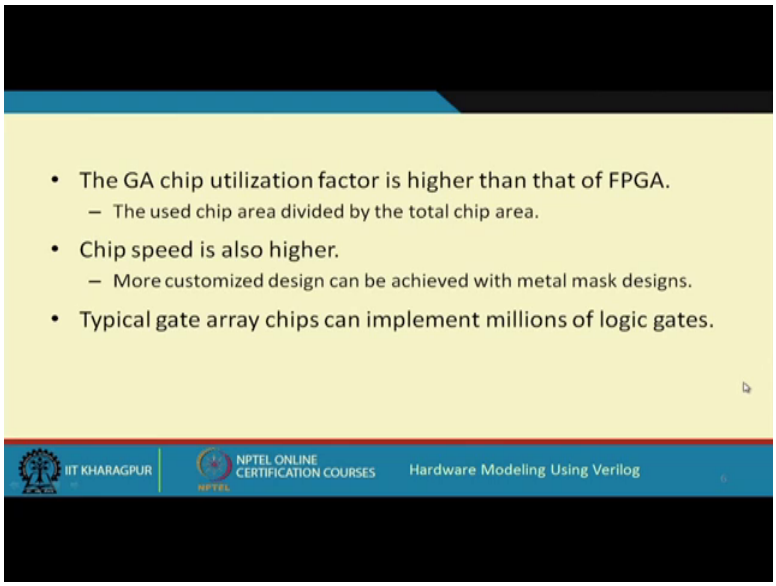
(Refer Slide Time: 04:53)



And again interconnect the gates to form our desired functionality. So, pictorially the process will look like this now in the first step which is shown on top. So, you simply create a large number of transistors using standard masks. So, what we get is a number of silicon wafers where the same large array of transistors is fabricated. Now you see when you are fabricating something in a fab, it will cost you money. So, when you are fabricating in millions your means; your total cost will be distributed or divided among the number of number of items you are manufacturing. So, because you are manufacturing these base wafers in large numbers the total cost of the first step is getting distributed among all these designs. So, that the per design cost of step one becomes very less.

Second step is the actual customization where you complete the metal interconnections. So, the transistors which are fabricated you interconnect them in some way, right. So, after this, you get your final design.

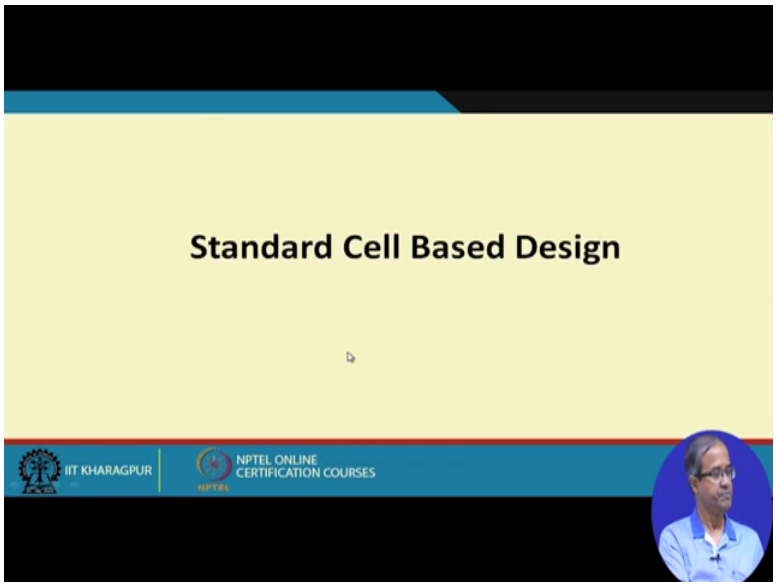
(Refer Slide Time: 06:13)



So, some features of this design style; number one because we are manipulating or working at the level of transistors clearly the chip utilization factor will be much higher as compare to FPGA because in FPGA chip in the field programmable gate array. If you recall, there was something called configuration logic block the CLBs, inside CLBs, there were the look up tables or LUTs. Now there apart from the LUTs, there were a large number of other circuitries as well lot of multiplexers some flip flops whether you need them or not those circuits are also there. So, may be out of that you are utilizing only 60 percent or 70 percent of the hardware the rest you are not actually using.

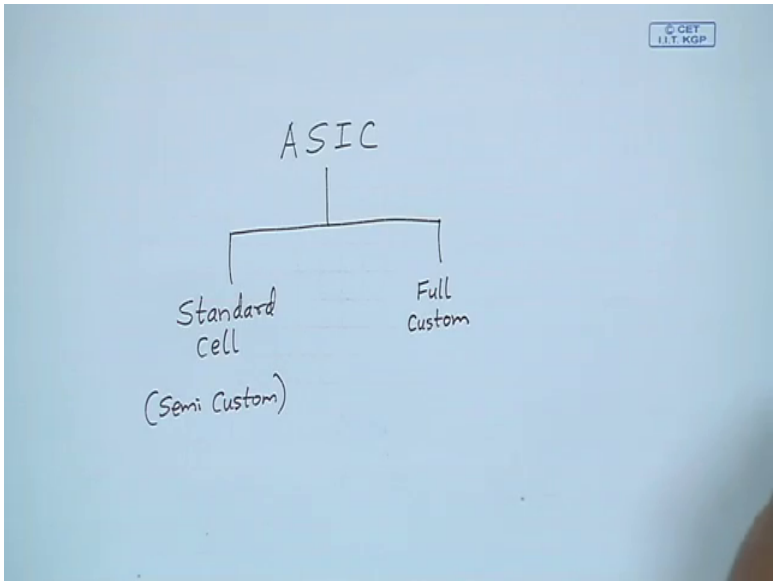
So, in that respect; when you go for this gate array kind of design style where you are working at a much lower level at the level of transistors. So, you can just utilize the transistors in a much better way. So, wastage will be very less here and because you are working at a much lower level, your chip speed in general will also be much higher because the delay will be less because all those you have all those complicated circuitry as there in FPGA they will not be there in gate array style. So, a typical gate array chip can implement millions of logic gates or even more. So, this is the idea.

(Refer Slide Time: 07:48)



So, next let us move on to a standard cell based design style. Now standard cell is one way to implement ASICs. So, under ASIC what we are trying to say is that.

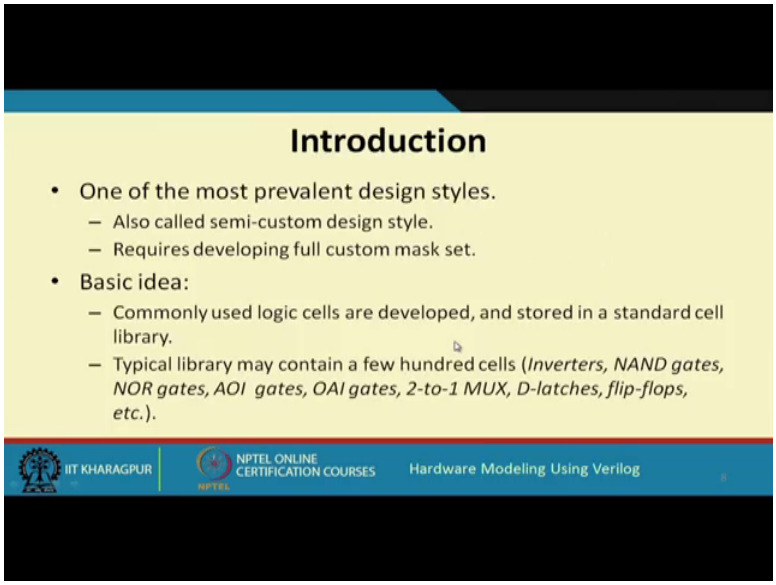
(Refer Slide Time: 08:05)



When we talk about application specific integrated circuit or ASIC, there are 2 broad ways of going about the manufacturing. One is called standard cell based, this is sometimes also called semi-custom and the other alternate style is full custom. These are the 2 broad approaches. Now in this approaches, the basic idea is that you have to go through all these steps of fabrication. So, your fabrication cost is not changing, right. So,

what is changing will be your design cost. Now in case of standard cell, you are using some pre designed cells from the library. You are picking them up and putting them in your design that is your standard sell or semi-custom design whereas, for full custom design you are theoretically designing everything from scratch. So, naturally it will take much long time to design a reasonably complex circuit.

(Refer Slide Time: 09:27)

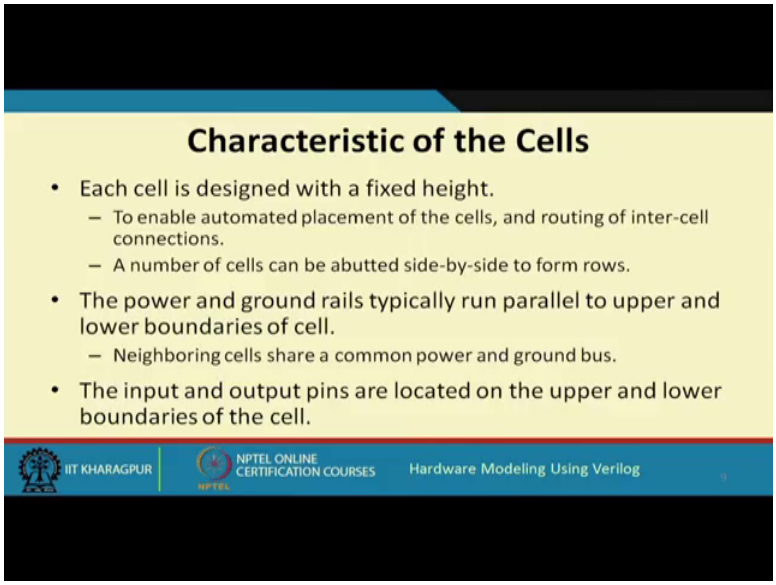


So, let us see the features of the standard cell based design style. So, this is one of the most widely used design style for manufacturing ASICs today. So, as I said, this is also called semi-custom because it is not full custom, lot of the design cells are already pre designed and available to us we are reusing them, but as I said, we have to go through all these steps of fabrication which means we have to develop the full set of masks that are required for fabrication. So, the basic ideas, I had said that you use some simple logic cells which are very commonly used and store them in a library which is typically called the standard cell library.

Now, when you store them in the library actually what we store is a highly optimized layout. So, it is not that it is a gate level net list we are taking the gate level net list putting it on our design and later on will be thinking about layout not exactly that we are getting the layout directly you can put it straight on to silicon that is idea. So, the typical library for standard cell may contain few 100 cells. So, cells may contain simple gates

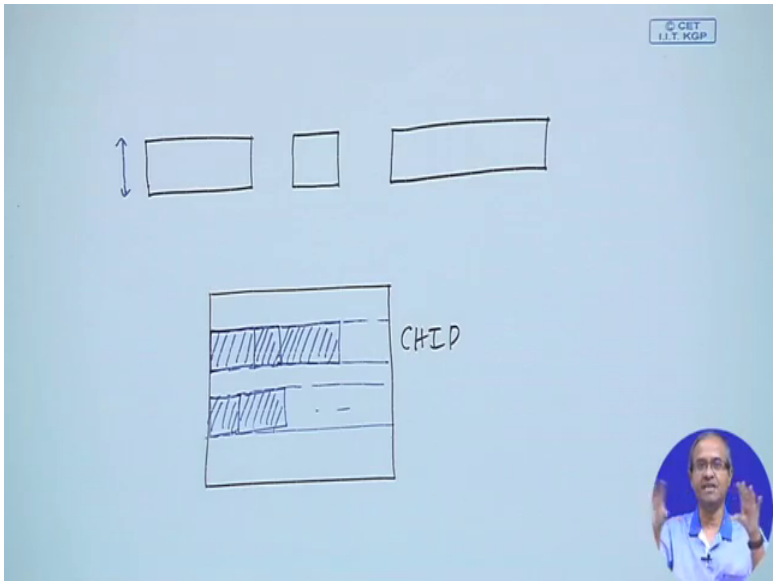
like inverters NAND, NOR or more complex gates like and or invert or and invert multiplexer latches flip flop, this kind of cells fine.

(Refer Slide Time: 11:03)



Now, there is something very unique about these cells, the cells are designed with a fixed height, but the widths may be different. So, the idea is as follows I am just trying to show you.

(Refer Slide Time: 11:21)

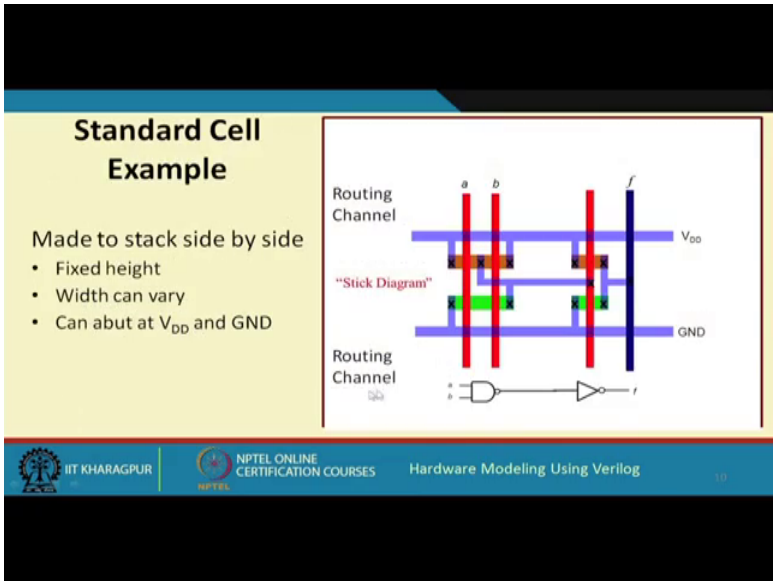


Suppose I have a cell like this which geometrically has a shape of this form, let us say I have another cell which has the same height.

But width is different, there is another cell same height, but the width is again different. So, when we put these cells on a chip suppose this is our final chip. So, here we place all these cells in some well defined rows, suppose this is one row, the height of a row will be equal to the height of this cell and within a row, what we do? We just place the cells one after the other, they will be physically touching each other right like this. So, you see the layout also becomes very regular all these cells will be arranged in terms of rows. So, some more cells will be placed here there be some cell here and so on. So, in standard cell, the layout looks something like this very regular where everything is placed in terms of rows. So, in terms of the design; in terms of the routing in terms of the floor planning the task of the designer becomes much simpler here, right.

So, just as I said, each cell is designed to the fixed height the power and ground rails the lines run parallel to the upper and lower boundary, I will just show an example, shortly the reason is that if you just connect 2 of the adjacent cells side by side, if you just connect them together the power lines VCC line will also VDD line will touch and also the ground line you will touch.

(Refer Slide Time: 13:35)

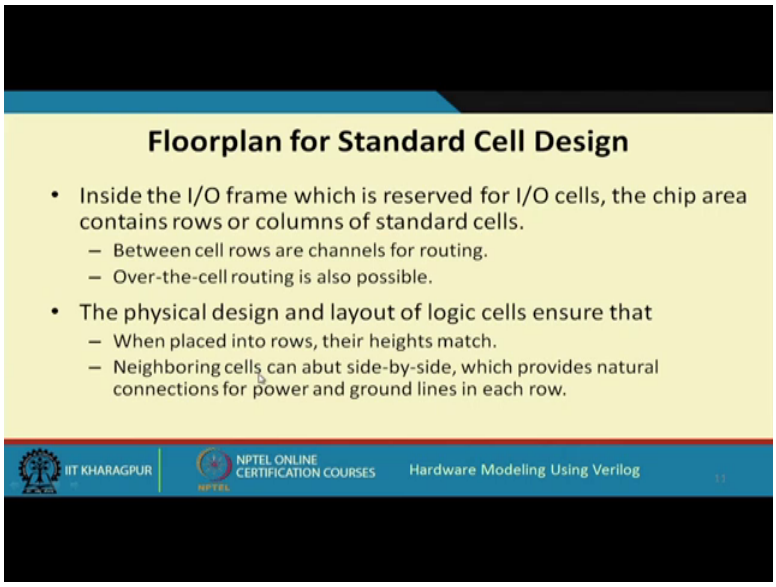


So, you will be getting continuous signal for the supplies power supplies and typically the input and output pins are look at on the upper and lower boundaries, let us take an example. So, here I am showing an example of a layout. Well, this is a way in which we show a layout. This is called a stick diagram. So, the blue lines are the metal connections

red lines are the poly silicon connections, green and brown are the diffusions, here are the n type transistor here the p type transistors and this black crosses are the this are interconnections. So, you see here we have implemented 2 gates; 1 NAND gate, 1one NOR gate, this part is the NAND gate this part is the NOR gate, the 2 inputs a, b.

So, you see the inputs a, b are available vertically here. So, the transistors that make this NAND gate, they are 2 transistors here and 2 transistors here and this is the output of this gate this output of the first gate feeds, the input of the second inverter there is an n transistor here, p transistor here. So, you see for both these, here we are showing that the 2 cells are touching each other. So, VDD is running on top ground is running on bottom. So, when they are brought together and touch each other, this VDD and ground lines become continuous. So, this is just an example fine.

(Refer Slide Time: 15:01)

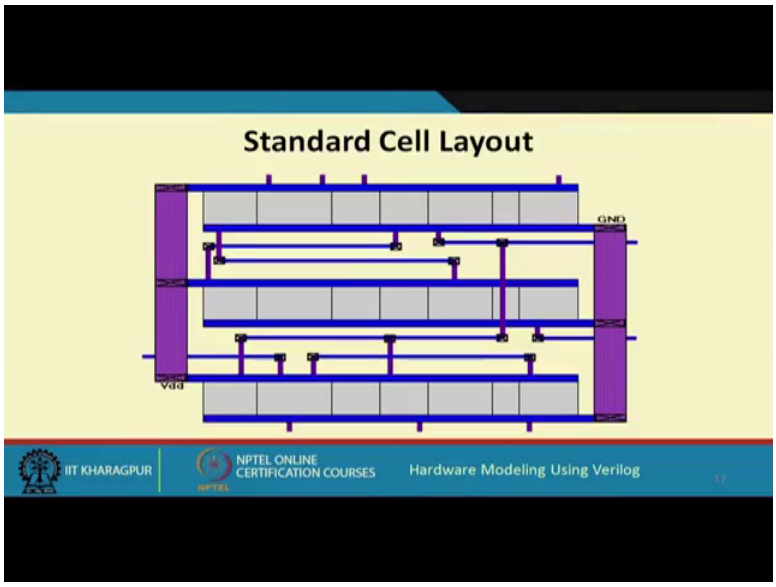


So, just as I had shown you. So, when you think of the floor plan of standard cells. So, the chip area actually will consist of some rows of standard cells or column whichever way you think of.

Now, the design and layout of the cells will; obviously, ensure that when you place them side by side their heights will match and in neighboring cells can abut side by side this is because of the restriction that you have imposed that all cells have to be of the same height because they are of the same height you can put the cells side by side and within a row, then any number of cells which can fit in a row you can put them side by side just

touching each other. So, the VDD will be on top ground will be on bottom and all the signal lines will be running vertically, they will be interconnected later on using some space which are available within the rows those are called channels.

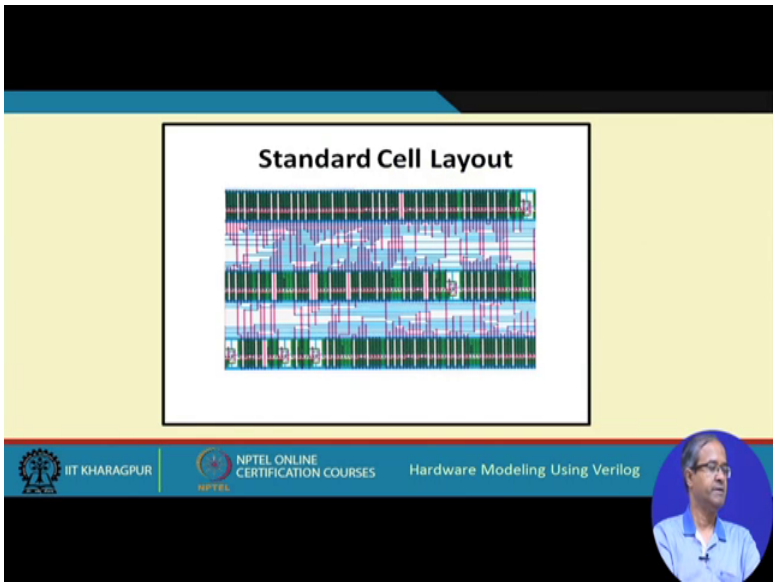
(Refer Slide Time: 16:09)



So, this is a typical standard cell layout which shows 3 rows; you see this is one row, this is one row this is one row.

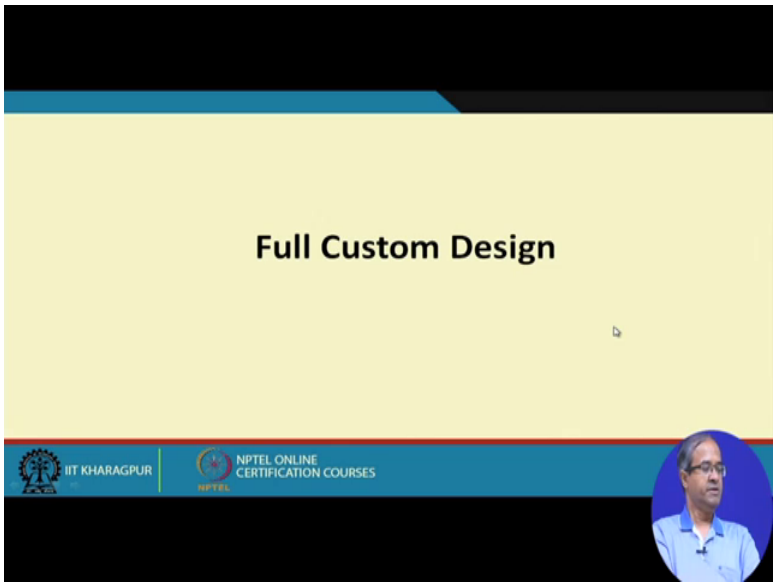
So, the different standard cells are shown, you see here; these are the cells and you see some of the interconnections are shown. So, this space between the rows, this part is called channel, you see some of the interconnections are made like this well it may so happen that one point here, a pin here wants to be connected or needs to be connected to a pin here. So, what do we do? We have to cross a row. So, there are some special cells also available which are called through cells. So, what it does? It simply provides the connection path from the top to bottom nothing else. So, you can use one of such cells to make a connection from here to here and then to here right and the power supply and the ground lines can be fed from 2 side. This is your VDD it feeds the VDD lines and the other side of ground it feeds the ground lines. So, this is a typical layout structure of a standard cell based design.

(Refer Slide Time: 17:30)



So, a typical standard cell layout for a moderately complex design is shown here, these are the cells and the interconnections you see they are quite complex right fine. Now let us come to full custom design.

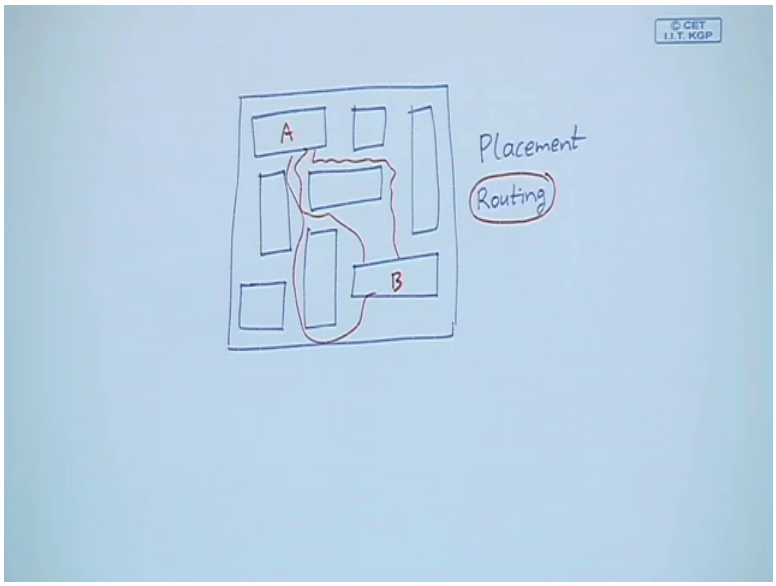
(Refer Slide Time: 17:46)



Now, for the standard cell or the semi-custom design, what we had assumed, we would assume that these cells are already available to us their layouts are available their geometries are very well defined their heights are same the relative vertical positions of the VDD and ground lines are the same. So, I can simply put the cells side by side, but in

full custom design we are not making any such assumption we are saying that our basic blocks in the design can be of any arbitrary complexity any arbitrary size some can be small some can be big you can place them in to the chip ire in whatever way you want like suppose this is your chip.

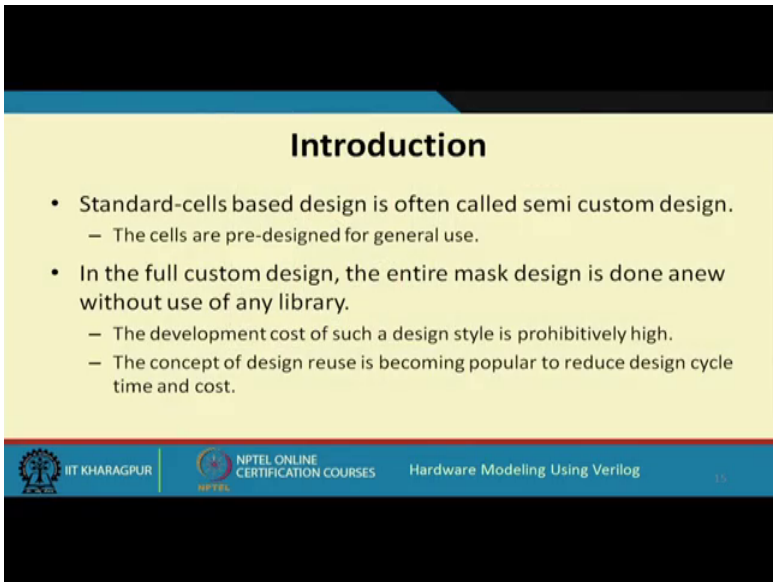
(Refer Slide Time: 18:37)



So, it is possible that one block is has a shape like this one block has a shape like this one has a shape like this one is like this one is like this. So, they can come in any arbitrary shapes and sizes.

So, you can see that here the problem is much more difficult here some very important problems pertain to placement of the blocks; how to place the blocks or in what way should I place the blocks. So, that my interconnection which is called routing becomes easier because say I may do one thing, I may place in such a similar in one way, let us say a block A and a block B, here they have a large number of connections between them. So, all these connections have to be brought either from here or from here or for some other direction. So, if you do not have sufficient space here for the connections you may see that your routing fails. So, if your routing fails you may have to go back you may have to modify the placement and again retry. So, this is not a very easy task, this is a quite difficult task, alright.

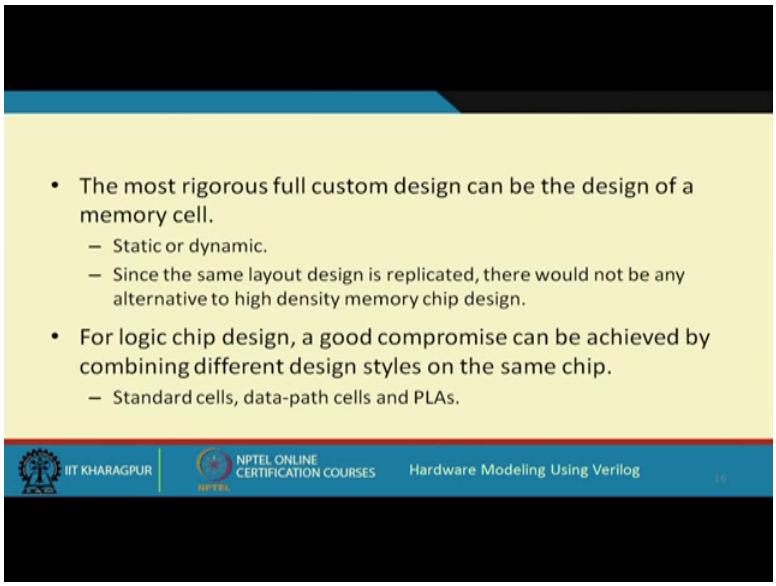
(Refer Slide Time: 19:58)



So, this standard cell is designed which you had seen this is called semi custom design semi because the cells that we are using those are pre designed, but the way we are placing the cells that is up to us, right, but in the full custom design we are theoretically will no one will do it this way today theoretically we are not using any library. So, the entire design is done from scratch, but; obviously, for the modern day VLSI chips where the number of transistors can run in to the in to the order of billions if you say that I will be doing everything from scratch. So, it really becomes impractical. So, it will take you years and years to design a particular chip, right. So, as I have mentioned here the development cost of such a design style may become prohibitively high therefore, all the designers today they use a concept called design reuse like I am just take an example.

Suppose you are coming up with a new is a new design of a processor she said I am developing a new CPU, I want to manufacture it fabricate it. So, should I design everything from scratch well, I will see that well I require a 32 bit fast multiplier which I had already designed earlier for some other processors. So, why not I take the same design from there and use it here. This is the concept of design reuse. So, today nobody designs everything from scratch whatever is available these are sometimes called IP codes intellectual property codes; they are taken from different sources they might be put together in a same place and you can integrate them in a suitable way fine. So, this is what is called design RIOs which is becoming very popular nowadays this; obviously, reduces the total time for design and also the cost they are related, fine.

(Refer Slide Time: 22:19)

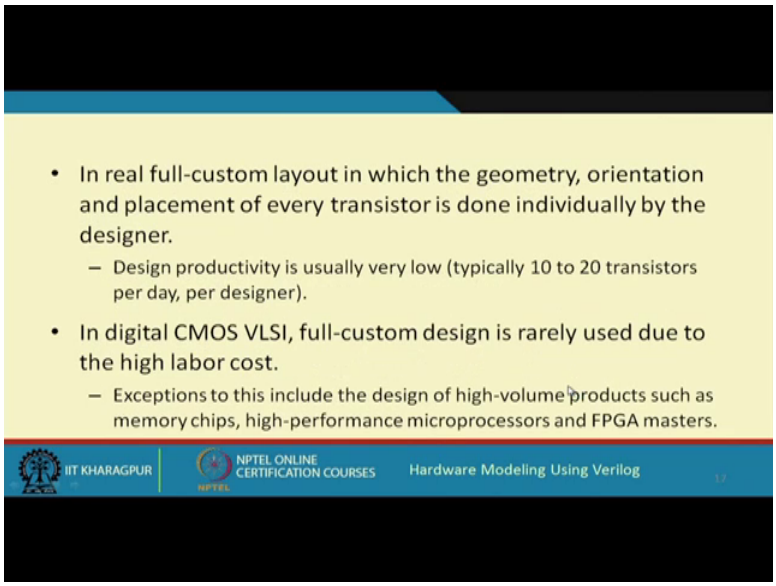


So, with respect to full custom design there is one kind of chip which always requires full custom design that is the memory cell because memory is the you can say the one design where you pack the maximum number of transistors in a chip well you look at 2 chip side by side one a processors one a memory chip in a memory chip because the layout of the transistors the way you are putting them are extremely regular and you have millions of such rows and columns you are using.

So, the design becomes very compact. So, you can put in much more number of transistors as compared to a processor design where your layout may not be that regular in a memory, there will be a large number of rows large number of columns with some transistors or some cells sitting at the junctions very regular kind of layout, right. So, memory is one such design where full custom design is religiously practiced. So, if you design one cell, the layout of one cell you can simply replicate it a billion times.

Let us say; so, it becomes easy, but for a design like a processors where there is a lot of logic then you can have a lot of mix and match some part you may implement in standard cell some part, you can have the blocks which are pre designed some parts you can use something called programmable logic arrays and so on.

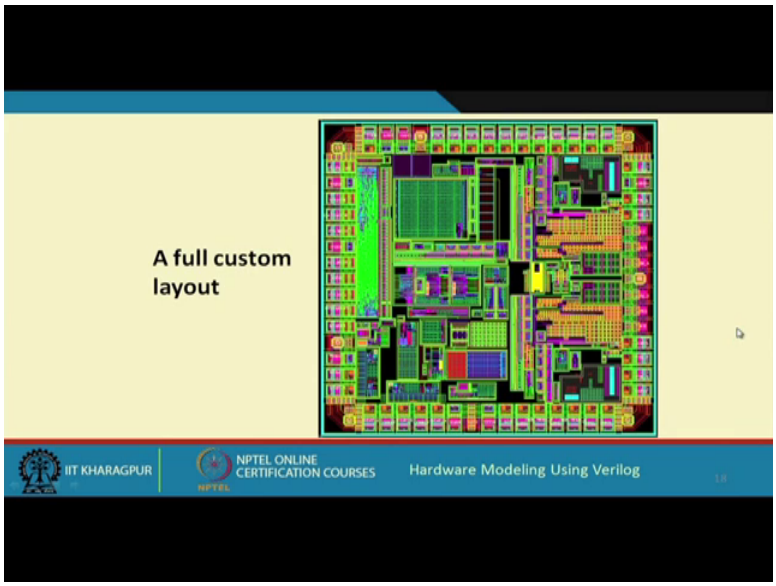
(Refer Slide Time: 24:06)



So, if you want to go for a full custom layout which is religiously your trying to start from scratch while orientation and placement of every transistor has to be done individually by the designer then you think. So, statistically it has been found that the design productivity for this kind of a design philosophy is typically 10 to 20 transistors per day per designer. So, even if a company puts in 1000 engineers, for this design you just imagine for a one billion transistor design how long will this take right because of the high labor cost and time this is actually never done.

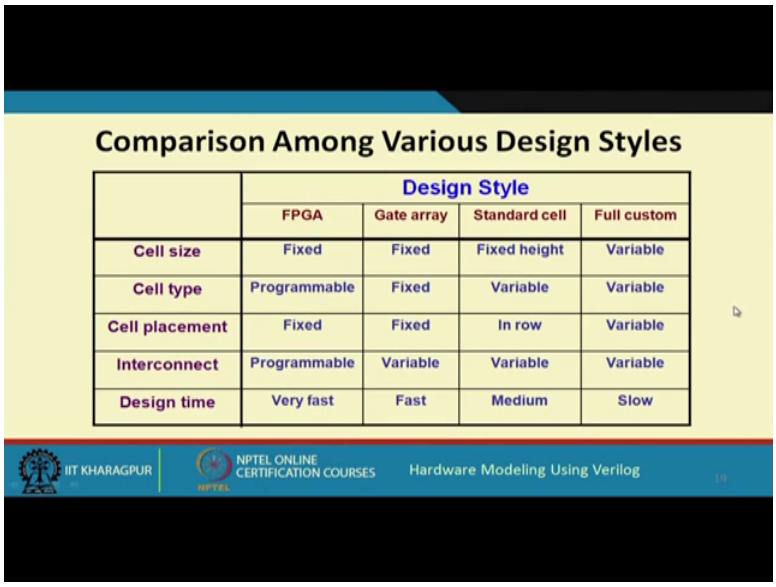
So, memory very high performance microprocessors like the one's INTEL manufacture for example, which are sold in millions and FPGA chips, the masters they are also sold in large quantities, these are some designs while you can put in some effort in order to improve their performance.

(Refer Slide Time: 25:19)



So, this is just an example of a full custom layout. So, the way it looks like you see there are different regions, they are all marked in different color sequencing this is one regular region, this is some region where there is not much regularity and so on.

(Refer Slide Time: 25:42)



	Design Style			
	FPGA	Gate array	Standard cell	Full custom
Cell size	Fixed	Fixed	Fixed height	Variable
Cell type	Programmable	Fixed	Variable	Variable
Cell placement	Fixed	Fixed	In row	Variable
Interconnect	Programmable	Variable	Variable	Variable
Design time	Very fast	Fast	Medium	Slow

These are the I O cells input output cells. So, just to compare among the design styles, we just now talked about FPGA gate array standard cell and full custom. So, we have ordered the designs in this way because FPGA is easiest full custom is the most difficult. So, we are comparing this with respect to these parameters cell size cell type placement

interconnect and the total time for design self-size in FPGA everything is prefabricated right this CLB, the LUTs they are fixed gate array also the transistors of the cells they are fixed standard cells; cells are not fixed, but the heights are fixed, but full custom you have complete flexibility they are variables.

Cell type in FPGA you have the LUTs, but you can implement anything by programming them. So, this is programmable, but in case of gate array there are some gate arrays where there are a large number of NAND gates which are built. So, the cell type is also fixed their NAND function, but standard cell there are various different kinds of cells full custom also there can be different kind of cells cell placement in FPGA and gate array are fixed because the cells are already there in standard cell you have some restriction cells can be placed along rows, but in full custom you can place them anywhere you want similarly for the interconnect for FPGA well again this is programmable, but for the other 3 styles interconnect has to be done based on the user requirement they are variable design time, FPGA is very fast full custom is the slowest gate array is here standard cell is here gate array we of course, faster than standard cell.

So, this is how the whole thing works. So, with this we come to the end of this lecture. Well, over the last 2 lectures; this one and the previous one, we have talked about the different design styles, well, most of you most of the time will probably be using the FPGA based design style only and there as the programming vehicle you will be using some harder description language like Verilog or VHDL as I had said there are other languages also available many people they also use those languages.

So, in our next following lectures, we shall now be moving in to the details of the very log language; some syntax some semantics and we shall be illustrating them with examples and we shall be progressing to more and more complex features and complex designs.

Thank you.